National Qualifications 2025

2025 Computing Science

Advanced Higher

Question Paper Finalised Marking Instructions

These marking instructions have been prepared by examination teams for use by SQA appointed markers when marking external course assessments.

General marking principles for Advanced Higher Computing Science

This information is provided to help you understand the general principles you must apply when marking candidate responses to questions in this paper. These principles must be read in conjunction with the detailed marking instructions, which identify the key features required in candidate responses.

(a) Marks for each candidate response must **always** be assigned in line with these general marking principles and the detailed marking instructions for this assessment.

(b) Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted.

(c) If a candidate response is not covered by either the principles or detailed marking instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.

(d) Award marks regardless of spelling, as long as the meaning is unambiguous. This applies to all responses, including code. Award marks as per the detailed marking instructions, regardless of syntax errors, if the intention of the coding is clear.

(e) For questions where candidates are asked to design or write code, a sample response is shown in the detailed marking instructions. This will not be the only valid response. You must use the detailed marking instructions and additional guidance to ensure that you consider alternative approaches and nuances of different programming languages. If in doubt you should refer to your team leader.

(f) If a candidate scores through a response and makes a further attempt, you should only mark the further attempt. If no further attempt is made and the original is legible, you should mark the original response.

(g) Where an incorrect response is carried forward and used correctly in a following part of the question, you should give credit for subsequent responses that are correct with regard to the original error. Candidates should not be penalised more than once for the same error.

(h) Only award marks for a valid response to the question asked. Where candidates are asked to:

- **Identify**, **name**, **give** or **state,** they need only name or present in brief form.
- **describe**, they must provide a statement or structure of characteristics and/or features. This will be more than an outline or a list. It may refer to, for example, a concept, process, experiment, situation, or facts, in the context of and appropriate to the question. Candidates must make the same number of factual/appropriate points as there are marks available in the question.
- **explain**, they must relate cause and/or effect and/or make relationships between things clear, in the context of the question or a specific area within the question.
- **write code,** they must write recognisable code, not prose nor a diagram.
- **design,** they must use a design technique appropriate to the problem. Award marks as per the detailed marking instructions, regardless of errors in the exemplification of the technique, if the intention of the design is clear.

(i) In the detailed marking instructions, if a word is underlined then it is essential; if a word is in brackets() then it is not essential. Words separated by / are alternatives.

Marking instructions for each question

Section 1 - Software design and development

Question			Expected response	Max mark	Additional guidance
1.			• a double linked list should be used • when a player lands on a move backwards square, they would need to traverse backwards in the linked list to the correct square; similarly, when they land on a move forward square, they need to traverse forward in the linked list	**2**	Award 1 mark for double linked list. Award 1 mark for justification. Accept doubly linked list.
2.	(a)		The company names are in alphabetical order.	**1**	Do not award mark for simply stating that data is in order; answer must clearly indicate that `companyName` field is ordered
	(b)		7 comparisons	**1**	
	(c)		• error 1 at Line 5. This should be low<=high to terminate the loop • error 2 at Line 6. This should be set middle value to (high + low)/2 • error 3 at Line 13. This should be set low to middle + 1	**3**	Award 1 mark for each error with correction indicated – do not award mark for simply stating the error. Accept error at Line 5 is high>=low

Question			Expected response	Max mark	Additional guidance
3.	(a)		• all orders received must be one of the three types: `SitIn`, `Delivery` or `Collection`, which each have a constructor method • there would never be a need to create an `Order` object	**1**	Award 1 mark for all orders being one of the subclass types. Accept also `Order` is an abstract class. Do not accept `Order` is superclass.
	(b)		• the `calcTotalOrder()` method in the `Delivery` class overrides the inherited method from the `Order` class • this is because the method in the `Delivery` class will add functionality to the inherited class to allow for the delivery charge to be added	**2**	Award 1 mark for use of override. Award 1 mark for additional functionality in context.
	(c)		• this is possible because of polymorphism • the array is of type `Order` which is a superclass and can store references to objects that belong to any of its subclasses since they are based on the `Order` class	**2**	Award 1 mark for polymorphism. Award 1 mark for subclass/ superclass connection.
	(d)		The error is because the `orderStatus` property is private due to encapsulation. A separate `getOrderStatus()` method would be needed.	**1**	Award 1 mark for private property that is encapsulated. Accept a (getter) method is needed to access the data.
	(e)	(i)	• although `order2` is an object of the `SitIn` class, in this situation, it is being indexed as an element of the `orderList` array which is of type `Order` • as the `Order` class does not have an `addExtraItem()` method, the subclass method is not accessible	**2**	Award 1 mark for class of `objectList[1]` being `Order` object not a `SitIn` object. Award 1 mark for `Order` class not being able to access `addExtraItem()` method belonging to the `SitIn` subclass.
		(ii)	Use the code `order2.addExtraItem()` to add an item to the order so that `objectList[1]` is updated.	**1**	Award 1 mark for description of a suitable solution.

Question			Expected response	Max mark	Additional guidance
3.	(f)		See sample solution below.	**4**	Award 1 mark for inner (fixed) and outer (conditional) loops correct (lines 2, 10 and 4 of the solution below). Award 1 mark for sorting in descending order indicating correct use of getters (line 5). Award 1 mark for swap objects (line 6). Award 1 mark for displaying top three after the sort, indicating correct use of getters (lines 11 & 12). Note: if candidate has sorted in ascending order and displays the last three elements in descending order, maximum of 4 marks can still be awarded.

```
1.   procedure bubbleSort(orderList[])
2.       start conditional loop
3.           set swapped = false
4.           start fixed loop from 0 to length of orderList[] – 2
5.               if orderList[loop].getOrderTotal < orderList[loop + 1].getOrderTotal then
6.                   swap the two order objects
7.                   set swapped = true
8.               end if
9.           end fixed loop
10.      end conditional loop when swapped = false
11       start fixed loop from 0 to 2
12.        display orderList[loop].getOrderNmber and orderList[loop].getOrderTotal
13.      end fixed loop
14.     end procedure
```

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>4.</td><td>(a)</td><td></td><td>DECLARE events INITIALLY
[[0.0] * 3] * 14</td><td>1</td><td>Award 1 mark for correct dimensions and type real indicated.

Accept valid declaration of 2D array with correct dimensions and data type in any programming language.</td></tr>
<tr><td></td><td>(b)</td><td>(i)</td><td>See sample solution below.</td><td>4</td><td>Award 1 mark for fixed loop from 2nd element to last element of array.

Award 1 mark for conditional loop with complex condition.

Award 1 mark for moving up entire row.

Award 1 mark for storing and inserting entire row.</td></tr>
<tr><td colspan="6">
<pre>
FOR outer FROM 1 TO 13   1st mark
  SET currentLat TO events[outer][0]
  SET currentLong TO events[outer][1]
  SET currentMag TO events[outer][2]
  SET inner TO outer
  WHILE (inner > 0) AND (events[inner-1][2] > currentMag) DO   2nd mark
     SET events[inner][0] TO events[inner-1][0]
     SET events[inner][1] TO events[inner-1][1]   3rd mark
     SET events[inner][2] TO events[inner-1][2]
     SET inner TO inner - 1
  END WHILE
  SET events[inner][0] TO currentLat
  SET events[inner][1] TO currentLong   4th mark
  SET events[inner][2] TO currentMag
END FOR
</pre>
</td></tr>
<tr><td></td><td></td><td>(ii)</td><td>Expected vs actual output:
The expected output from the sort code would be predicted and noted. After the sort has been executed, the contents of the sorted array could be displayed on the screen. A screenshot of this display could be compared with the predicted results to determine whether the sort code is working correctly.

Use of breakpoint:
The contents of the <code>events</code> array could be inspected (using a break point) before the sort code is executed and then inspected again after it been executed. Screenshots of the before and after contents of the arrays could be presented as evidence to show that the sort code is working correctly.</td><td>2</td><td>Award 1 mark for inspecting contents of array before sorting.

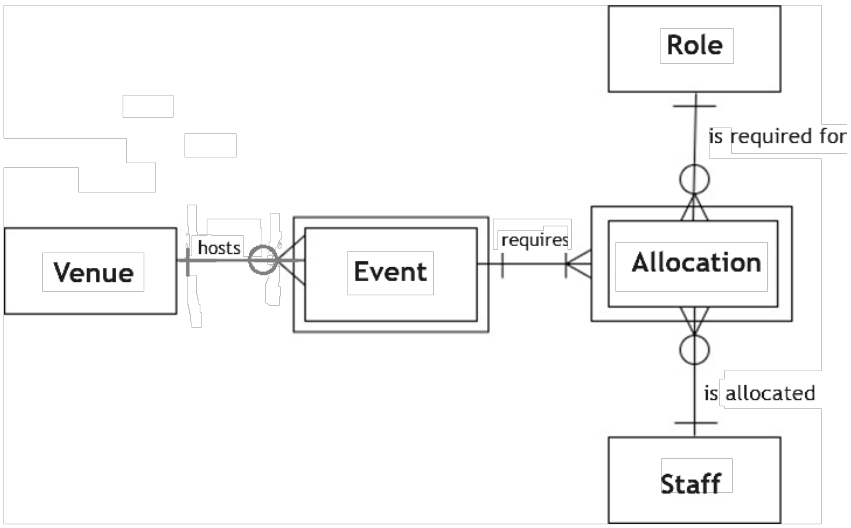
Award 1 mark for outputting/inspecting data after sorting to compare with expected output.</td></tr>
</table>

Question			Expected response	Max mark	Additional guidance
4.	(c)		See sample solution below.	**4**	Award 1 mark for nested fixed loop (lines 1 & 5 of the solution below). Award 1 mark for comparison of latitude and longitude differences (line 6). Award 1 mark for count that doesn't count event itself (line 6). Award 1 mark for updating cluster count in clusterSize array (line 7).

```
1.  start fixed outer loop from 0 to 13
2.      set clusterSize[outer] = 0
3.      set currentLat = events[outer][0]
4.      set currentLong = events[outer][1]
5.      start fixed inner loop from 0 to 13
6.          if (currentLat – events[inner][0] between 5 and –5 inclusive) and (currentLong –
            events[inner][1] between 5 and –5 inclusive) and (outer ≠ inner) then
7.              set clusterSize[outer] = clusterSize[outer] + 1
8.          end if
9.      end fixed loop
10.  end fixed loop
```

	(d)	(i)	• perfective maintenance is needed **Justification:** • the software has been running successfully but new functionality is required to record volcanic activity • the code is being updated to run in an OOP environment by altering the internal structure of existing code using 2D arrays to run with methods and objects	**2**	Award 1 mark for perfective maintenance. Award 1 mark for suitable justification. Note: only 1 justification is needed. Note: Do not accept adaptive as code is not being updated to take account of external factors.
		(ii)	• economic feasibility: Check whether there is a financial benefit in spending money to create a new OO version of the software; if there is no financial gain, the project would not be feasible • technical feasibility: Check whether the skills necessary to develop an OO solution are available amongst the developers otherwise the project would not be feasible	**2**	Award 1 mark for explanation of the need to consider economic feasibility. Award 1 mark for explanation of the need to consider technical feasibility.

Section 2 - Database design and development

Question			Expected response	Max mark	Additional guidance
5.			Results from a user survey could be used to: • identify end user requirements • identify functional requirements • identify requirements spec • identify resources to be used • create Use Case diagram	**2**	Award 1 mark for each bullet Maximum 2 marks. Note: since the requirements spec includes the end-user and functional requirements, a maximum of 1 mark should be awarded if either one is suggested along with requirements spec. Note: Candidates that do not name items from the expected response but describe specific and relevant examples of requirements can be awarded 1 mark.
6.	(a)		See sample solution below.  	**4**	Award 1 mark for mandatory-to-mandatory relationship participation between Event and Allocation entities. Award 1 mark for all 3 mandatory-to-optional relationship participation between Venue-Event, Role-Allocation and Staff-Allocation. Award 1 mark for all 3 mandatory-to-optional relationships found the right way/in the right order. Award 1 mark for Allocation and Event indicated as weak entities.

Question			Expected response	Max mark	Additional guidance
6.	(b)		• use of a surrogate key makes it easier to reference historical data • use of a surrogate key improves the performance of queries that involve a join with the Event, Role or Staff tables	**1**	Award 1 mark for one correct reason. Do not accept quicker/easier without supporting statement.

	(c)		See sample solution below.
			
			`PRIMARY KEY(eventID)`
			`FOREIGN KEY(venueID) REFERENCES Venue(venueID)`
				2	Award 1 mark for primary key.
					Award 1 mark for foreign key.

7.	(a)		See sample solution below.
			
			`date BETWEEN '2025-03-01' AND '2025-03-31'`
			`HAVING SUM(salePrice) < targetSales`
				2	Award 1 mark for correct use of `BETWEEN`
					Do not accept `LIKE` as it is not an AH concept.
					
					Award 1 mark for correct use of `HAVING`
					Note: ignore incorrect date formats.

	(b)		A: `salePrice > ANY` B: `make IN('Goudi', 'CNX')`	**2**	Award 1 mark for correct use of `ANY` Award 1 mark for correct use of `IN`
	(c)		• the subquery generates a list of customers who purchased environmentally friendly cars in 2023 or 2024 • `NOT EXISTS` ensures that only customers NOT found in the subquery appear in the query answer table. If a customer is absent from the subquery they will be included in the final output	**2**	Award 1 mark for an explanation of how the sub query generates initial results. Award 1 mark for an explanation of how `NOT EXISTS` is used in the outer query.

Question			Expected response	Max mark	Additional guidance
7.	(d)	(i)	See sample solution below. A: `method = "POST"` B: `<input type = "submit" value = "Submit">` **OR** B: `<input type = "submit">` **OR** B: `<input type="button">Submit` **OR** B: `<button>Submit</button>`	**2**	Award 1 mark for correct use of `POST`. Award 1 mark for input type and value. Note: responses with type = "submit" `value = "Submit"` is not required; the alternative response `type = "button"` and `<button>` both require Submit.
		(ii)	• integrative testing is needed for projects that require integration of separate components • in this situation, integrative testing is needed to verify data is passed correctly between the website and database	**2**	Award 1 mark for description of integrative testing. Award 1 mark for data connection between website and database.
		(iii)	End-user testing would be conducted at the end of the development, with the end users of the software using a live version of the solution and providing feedback to the developers.	**1**	Award 1 mark for description that indicates role of end-users in providing feedback to the developers.

Section 3 – Web design and development

Question			Expected response	Max mark	Additional guidance
8.			Results from a user survey could be used to: • identify end user requirements • identify functional requirements • identify requirements spec • identify resources to be used • create Use Case diagram	**2**	Award 1 mark for each bullet Maximum 2 marks. Note: since the requirements spec includes the end-user and functional requirements, a maximum of 1 mark should be awarded if either one is suggested along with requirements spec. Note: Candidates that do not name items from the expected response but describe specific and relevant examples of requirements can be awarded 1 mark.

Question			Expected response	Max mark	Additional guidance
9.	(a)		**Media print** • use `@media print` instead of media screen **Additional changes** • change the background colour from lightgrey to white (or none) • remove the second image • reduce the size of the font and/or alignment used to display the heading	**2**	Award 1 mark for use of media print. Award 1 mark for any two additional changes to produce the paper output. A minimum of two must be indicated. Accept correct code as alternative to description. See sample solution below.

			`@media print {`
			`body {background-color: white}`
			`.flower1img {display: none}`
			`.flower3img {width: 150px; height: 150px}`
			`.flower2img {display: none}`
			`.heading {font-family: Lucida Handwriting; font-size:12px; text-align: right}`
			`}`		

	(b)		• integrative testing is needed for projects that require integration of separate components • in this situation, integrative testing is needed to verify data is passed correctly between the website and database	**2**	Award 1 mark for description of integrative testing. Award 1 mark for data connection between website and database.
	(c)		End-user testing would be conducted at the end of the development, with the end users of the software using a live version of the solution and providing feedback to the developers.	**1**	Award 1 mark for description that indicates role of end-users.

Question			Expected response	Max mark	Additional guidance
10.	(a)	(i)	• session variables allow values stored in variables to persist across several pages of the website. In this situation, session variables are needed to store the details submitted by a competitor that they can shared across several pages of the website so that the details can be stored in the database using code on the `saveDetails.php page` • values submitted by the user on page `signup.html` are `POST`ed to `confirmation.php`. A session must be started on the `confirmation.php` page to initialise the session variables using the submitted values • A session will also be needed on the `saveDetails.php` page to ensure that the values contained in the session variables are available and can be accessed by the code used to save the competitor's details to the database	**3**	Award 1 mark each for any 3 of the following: • description of need use made of session variables to share data across several pages of a website • need for a session on the page `confirmation.php` to initialise session variables • use made of submitted values to initialise the session variables • need for a session on the `saveDetails.php` so that can values stored in the session variables can be accessed Maximum 3 marks

		(ii)	See sample solution below.
			
			`$server = "jigsaw1";`
			`$username = "admin";`
			`$password = "pass1234";`
			`$database = "jigsawFixtures";`
			
			`$link = mysqli_connect($server,$username,$password,$database);`
				2	Award 1 mark for correct assignment use of details provided in the question. This may be indicated as separate assignment statements or as parameter values of the `mysqli_connect()` function.
					
					Award 1 mark for correct use of `mysqli_connect()` function in connection code with the 4 correct parameters or assignment details in the correct sequence.

Question			Expected response	Max mark	Additional guidance
10.	(b)	(i)	An ascending sort is used to arrange the competitor details from shortest to longest time taken to complete the round 1 jigsaw puzzle. This is needed to ensure that the best 24 times are listed first.	**1**	Award 1 mark for explanation of importance of ascending sort.
		(ii)	• the `mysqli_fetch_array()` function is used to allocate a single set of results returned by the query to the `$row` variable and will terminate the loop when there are no more results to be processed • the `$numProcessed` variable is incremented at the end of each iteration and will terminate the loop once the first 24 sets of competitor details have been processed	**2**	Award 1 mark for explanation of how `mysqli_fetch_array()` function is used to terminate the loop. Award 1 mark for explanation of how `$numProcessed` is used to terminate the loop.
		(iii)	1. if $numProcessed = 0 then 2. set $points = 8 3. elseif ($numProcessed = 1) 4. set $points = 5 5. elseif ($numProcessed = 2) 6. set $points = 3 7. else 8. set $points = 1 9. end if	**2**	Award 1 mark for use of `$numProcessed` variable to make comparison. Award 1 mark for updating points correctly.

		(iv)	See sample solution below.
			
			`echo '<tr><td>'.$row['forename'].'</td>'." ".`
			`'<td>'.$row['surname'].`
			`'</td><td>'.$row['timeTakenRound1'].`
			`'</td><td>'.$points.'</td></tr>';`
				2	Award 1 mark for use of `echo` with `forename, surname, timeTakenRound1` and `$points`.
					
					Award 1 mark for correct use of `<tr>`, `</tr>`, `<td>` and `</td>`
					
					Note: do not penalise for incorrect or missing `$row`, ' or "

		(v)	See sample solution below.
			
			`UPDATE Competitor`
			`SET competitorPoints = '$points'`
				1	Note: do not penalise for missing inverted commas.

[END OF MARKING INSTRUCTIONS]